Bruce Xi

(236)-777-8218 • brucexi99@outlook.com • bruce-shidi-xi • brucexi999

Education

The University of British Columbia **Vancouver, BC**

Master of Engineering in Electrical and Computer Engineering, GPA 4.0 *2021–Present*

- Relevant courses: Digital/Microcomputer System Design, Computer Architecture, VLSI, IC Testing and Reliability, DNN Accelerator, Deep Learning

Imperial College London **London, UK**

Bachelor of Engineering in Materials Science and Engineering, First-Class Honours Degree *2018–2021*

- Obtained Dean's List for three consecutive years (2018-2021)

Experience

Motorola Solutions **Vancouver, BC**

Design Validation Co-op *May-Dec. 2022*

- Conducted extensive surveillance camera tests, validating the cameras' electrical, mechanical, and optical performance
- Engineered Python-based software for test automation and data analysis, enhancing test efficiency with up to 90% automation
- Utilized FFmpeg for video processing and analysis
- Collaborated within a team using Git, GitHub, Confluence, and Jira to optimize workflow efficiency

Project

Design, Simulation, and Analysis of a 10 Gbps Differential SerDes Data Link *Jan.-May 2024*

- Designed a 10 Gbps differential SerDes data link using Cadence Virtuoso in 45 nm technology, incorporating a TX, a differential channel, and an RX. Performed circuit simulations using Spectre, achieving a power consumption of 9 mW, an eye-opening of 47 mV under the worst-case data pattern, and zero errors with PRBS7 input over 10,000 UIs.
- Characterized channel impulse and pulse responses using MATLAB, determining the worst-case data pattern and computing tap values for 2-tap FIR equalization.
- Developed the TX with a 4:1 serializer based on a 5-latch-2:1 multiplexer topology with half-rate clocking, requiring 5 GHz clock frequency at max. The pre-drivers are designed as inverter chains optimized using logical effort. Implemented a high-swing, voltage-mode driver segmented into 10 parts to provide a tunable impedance of $50\Omega \pm 30\%$ and configured to pre-emphasize the signal, functioning as an FIR filter.
- Implemented the RX with 1:4 deserialization at 2.5 GHz using four slicers followed by synchronizers. Each slicer integrates a track-and-hold switch executed as a transmission gate, a StrongArm latch, and an SR latch.

AMBA AXI Stream Header Insertion **Feb.-Mar. 2024**

- Designed a Verilog RTL module for inserting headers into network packets, adhering to the AXI Stream protocol. The module receives headers and data packets from two master interfaces and sends them to one slave after packing and merging
- Implemented pipelining for both data and handshake signals, incorporating skid buffers to eliminate bubbles and efficiently manage backpressure

Concurrent VLSI Routing with Multi-agent Deep Reinforcement Learning **May-Oct. 2023**

- Developed a Python-based machine learning framework to address the VLSI global routing problem in a concurrent manner. This framework modeled routing as a pathfinding task and solved it using multi-agent reinforcement learning integrated with deep neural networks (MLP and GNN implemented in PyTorch)
- Addressed training challenges by fine-tuning hyperparameters using grid search, leading to significant performance improvements
- The proposed work overcame the traditional net-ordering issue, guaranteed zero overflow, and outperformed an A* baseline by 2.6% in terms of wirelength

Microcomputer System Development **Jan.-Apr. 2023**

- Engineered key components of a microcomputer system including a 4-way set-associative cache and a DRAM controller using Verilog. The cache reduced the runtime of a benchmark by 43%
- Developed software in C that controlled a soft Motorola 68000 CPU to communicate with peripheral hardware (Flash, EEPROM, and ADC/DAC) using SPI, IIC, and CAN protocol
- Utilized hardware timer interrupt and designed a snake game in C that ran on the system and interact with the player using VGA

CPU Design and Assembly programming **June-Sept. 2022**

- Architected a 16-bit RISC CPU from the ground up using Verilog, integrating pivotal components such as FSM, datapath, RAM, and I/O interfaces. The CPU was implemented on an Altera FPGA
- The CPU supported 13 diverse instructions encompassing ALU operations, memory access, and branching mechanisms
- Implemented preemptive multitasking on an ARM core integrated on the FPGA using ARM Assembly

FPGA SoC Design **Jan.-Apr. 2022**

- Developed a comprehensive SoC on FPGA using Quartus, incorporating a Nios II soft processor, on-chip RAM, and custom IP cores, managed via Avalon memory-mapped interface
- Designed and synthesized custom IP cores in Verilog for functionalities like image data processing, arithmetic acceleration and VGA output
- Created software in C to perform system tasks such as performance benchmarking and control over VGA display

Skills

Hardware: Verilog, SystemVerilog, FPGA, ModelSim, Quartus, Cadence

Software: Python $>$ C $>$ ARM Assembly $=$ C++, Linux, MATLAB